INFORMATION MANAGEMENT SERVICES DIVISION

NEBRASKA DEPARTMENT OF ADMINISTRATIVE SERVICES

DATA BASE MANAGEMENT

DB2 APPLICATIONS DEVELOPMENT REFERENCE GUIDE

Prepared for: State of Nebraska

Information Management Services

by: DBM STAFF

Document No.: 95089CDP0248

FOREWORD

In the traditional development environment, projects are often started at the analysis or design levels. All too many times, these projects are begun to satisfy the perceived need to have a file, which contains certain information, or to have a program, which can accomplish certain tasks.

The problem with projects which start with these objectives has been the failure to fully address or to fully communicate the business requirements or needs which are behind the development requests. Assumptions about the client's business needs, or assumptions about the final capabilities of a software program, often lead to the development of systems which take longer to develop than is initially estimated, which costs more than is originally budgeted, and which do not provide the full information or functionality that is originally envisioned by both the client and the developer.

To address these shortcomings with traditional development, methodologies have been defined which provide a way of defining, scoping, and accomplishing projects to best meet the business requirements of the client areas. While the methodology provides a guide to the development process, and while CASE tools are available to help developers progress through the methodology processes, there is still a need for standards and procedures to define the specific development requirements of the individual data processing environment.

Standards and procedures provide a framework of actions and methods by which both project development and communication are universally accomplished. This framework provides both a list of deliverables for the development staff to accomplish, as well as, a list of expectations upon which all interested parties may act. While many of these tasks may seem trivial, and others may seem cumbersome or involved, each contributes to a more complete project plan, better communication and planning, and, therefore, to the more likely chance of project success.

Information Management Services (IM Services) and the many agencies that IM Services serves do not have universal standards and procedures. Rather, development standards and procedures are generally maintained at the agency level, while performance standards and operational procedures are maintained at the IM Services level.

IM Services defines standards and procedures in the following way:

STANDARD

"A standard is a yard stick against which any task can be measured to assure it has been completed using the correct steps and procedures."

PROCEDURE

"A description of the steps to be followed, in a regular and specified order, to carry out the tasks required by a standard or those recommended in a guideline."

The following DB2 Application Development reference information is not meant to be all-inclusive. It is intended to be used as a framework for DB2 applications development as that development relates to an operation within IM Services' central data base management systems. The guidelines and information are provided in anticipation of an expected set of events, and an expected set of behaviors which help all persons involved in a development effort to know what is required at all phases of the development cycle.

The following goals and operation actions are presented as the minimum framework to be used for both traditional and IE project development efforts. No effort has been made, however, to declare this information as fully comprehensive or complete Where additional goals or operation actions are found to be desired or required, they will be added to the document. Where they need to be changed, changes will be considered.

PARTICIPATION IN PHASES OF DEVELOPMENT

OBJECTIVE: Data Base Management will participate in all development phases of a project.

1. Data Base Management will assign an appropriate number of trained and experienced staff to a project to provide data base knowledge and support to that project.

2. Project Management will schedule projects, which involve Data Base Management staff, with sufficient lead-time to arrange staff schedules as required.

3. Senior IM Services management will be the final authority regarding project priorities and/or scheduling conflicts.

4. Conflicts in priorities or scheduling, will not be reason to circumvent or ignore any of the standards or procedures presented in this document.

FEASIBILITY

(Also Planning)

OBJECTIVE: Data Base Management will provide consultation services in the feasibility or planning phase of a project.

1. All projects must be evaluated based upon their importance or their payback in the framework of the Enterprise model. The feasibility of a project is normally determined within the dictates of the Integrated Strategic Plan, and should include the reason or the payback, which gives the project its importance.

2. Traditional projects typically are based upon a more direct and immediate payback from the project itself, rather than on an integrated benefit derived from its importance to the Enterprise. The Data Base Management group will participate in this phase and will provide good faith estimates for development time and effort based upon available information. These estimates can and should be used in preparing project justification.

3. In the event that business or regulatory requirements dictate the initiation of a project, without payback or cost justification, the Data Base Management group should be consulted to accomplish the following two objectives:

a. The Data Base Management group will have the information necessary to arrange schedule-time and manpower availability to support the project.

b. The Data Base Management group can provide a gross estimate of time and manpower needed to accomplish data base related tasks.

BUSINESS AREA ANALYSIS

OBJECTIVE: At least one Data Base Management staff person will be attached to a project during the Business Area Analysis phase and will review all deliverables from this phase.

1. In the Information Engineering Methodology (IEM), the Business Area Analysis part of a project is completed in one phase (small and traditional projects), or in two phases in larger projects (Outline and Detailed BAA's). This phase defines the business requirements and rules, which will dictate the necessity for the project. It is this area analysis around which the system is designed.

2. The Data Base Management group will attach at least one person to the project during this phase. The DBM will be included in all 'reviews' conducted in this phase, which will ensure a complete understanding of the business requirements being addressed. This phase participation will further serve the effort to understand the requirements of the project, as well as to more fully appreciate the volumes and/or criticality of processes and data, which will ultimately be included in the construction of the project.

3. The Analysis project team will produce the following deliverables for DBM review:

- A complete Entity Relationship Diagram (ERD). This diagram will normally show all entity types and relationships in third normal form. Denormalization for performance will be addressed during the design phase. All 'Many to Many' relationships should be resolved to facilitate the naming of resultant associative entity types.

- Include with the ERD descriptions of entities and attributes. An example of VENDOR entity would be 1 and only 1 VENDOR can have 1 to M vehicles.

- When possible, Process Logic Diagrams (PLD's) should be created. These diagrams help depict the Process and Data interactions of the business. While the formal methodology suggests that individual Process Logic Diagrams be prepared for each elementary process, grouped notations on a copy of the ERD will, in many cases, satisfy the intent of this documentation.

4. In all cases, physical DB2 objects will be created only after the DBM group has reviewed the objects and has had sufficient opportunity to understand the data and relationships in the system or subject area. The review will normally be conducted by the DBM staff member attached to the project. This presentation will help insure that the staff member has sufficient understanding of the system or subject area and that the documentation supports the facts as they are presented. Under normal circumstances, no more that twenty (20) tables and relationships will be presented at one time, and no more than 2 reviews will be conducted during a one week time period.

BUSINESS SYSTEM DESIGN

OBJECTIVE: At least one Data Base Management staff person will be assigned to a project during the Business System Design phase. The DBM staff member will review all models and diagrams and gain a thorough understanding of the proposed system processing and the evolving data base objects.

1. Data Base Management will provide the project with the technical design of its data base system.

- A staff member will be assigned to the project.

- The entities will be translated to DB2 DDL. This phase will mark the first intensive participation by the DBM group.

- The Analysis project team will use the design tools to remove errors and will generate the DDL to be ported to the mainframe.

- Errors found during this phase will be corrected by the design team for correction. If errors are substantial, the translation process will be repeated after the error correction process.

2. The physical objects defined in the CASE environment will be a test project database, tablespaces, tables, and all indexes. The design team will provide the DDL along with the following physical design estimates:

Initial volume and growth of each entity:

a. Estimate of number of rows for each table

b. Growth statistics (number of rows that will be added in the future).

c. Growth horizon (how often the amount of "Growth Statistics" will be added.

i.e. 2,000 rows added every year

CONSTRUCTION

OBJECTIVE: At least one Data Base Management staff person will be assigned to a project during the Construction phase. The DBM staff member will provide technical assistance to the development staff in both their programming efforts and in their testing environments.

1. In general, all program SQL and DML will be designed and coded in the most resource efficient manner possible.

2. No application program will be designed nor coded using dynamic SQL, unless reviewed and approved by Data Base Management.

3. Individual program SQL code may be subject to review and approval by Data Base Management.

4. At the discretion of Data Base Management, the program may be subject to monitor and/or review in a CAT/Volume environment to ensure program execution efficiency.

5. All batch programs will use checkpoints or commit points in their processing with the following exceptions:

- Read only programs, which are estimated to run less than 15 minutes.

- Update programs, which are estimated to run less than 30 seconds.

- One-time, request, or conversion programs which are estimated to run less than 30 minutes, and for which an image copy is taken prior to the run.

- Any program, which for business or operational considerations, is approved by Data Base Management to run without issuing checkpoints and/or commit points.

6. All batch programs will contain appropriate code to ensure that they can be restarted from the last checkpoint or commit point.

7. Data Base Management will promote all DB2 objects to the appropriate test levels.

8. Project Analysis team will be responsible for arranging the loading, restoring, or copying of data in all test environments.

9. All data used in a test environment must be either:

- created in the test environment based upon fictitious persons, conditions, or combinations of data which do not resemble nor can be used to identify real persons or conditions in a department's production data base.

- production data which has been sufficiently sanitized or altered to prevent the ready identification of persons or conditions by others not associated with the department nor its business.

DB2 OBJECT NAMING CONVENTIONS

OBJECTIVE: All DB2 data base objects and all DB2 object qualifiers or creators will use a standard naming convention.

DB2 FILE STRUCTURE

DB2 tablespaces and index spaces use VSAM for the access method and are created as data sets with the following name format:

sss.DSNDBD.ssstDBnn.sssTSnnn.I0001.A001

- sss = VCAT Alias name
- All Test = DBT
- Production = System Acronym + '2' (e.g. RFS2, NAS2, LCC2)
- DSNDBC = DB2 Cluster designation
- ssstDBnn = Data Base name (see description below)
- sssTSnnn = Table space (see description below)

or

- iiiiiii = Index space (generated by DB2)
- I0001 = literal
- Annn = Partition (A + partition number)

ACCESS

In production, RACF ALTER access must be granted to the following system and support groups based on the two high nodes of the DB2 data sets (i.e. sss2.DSNDBC.*.**):

- CDPPROD
- DB2MSTR
- DB2DBM1
- DBM0000

NOTE: if using a STOGROUP and not SYSDEFLT also request DBTMSTR, DBT2MSTR, DBTDBM1 and DBT2DBM1

STORAGE GROUPS

1. The default DB2 Storage Group SYSDEFLT will normally be used in the test system.

2. Special DB2 Storage Group(s) may be set up in test when the volume of test objects or charge-back considerations warrant their use.

3. DB2 Storage Groups will be used in the production system and will be named using the following naming convention:

xxxSTOGn where:

- xxx = Department (System) Acronym
- STOG = Literal
- n = Number

DB2 DATA BASE

1. The default DB2 database DSNDB04 will not be used by any application project.

2. Project data bases will be named using the following naming convention:

xxxtDBnn where:

- xxx = Department (System) Acronym
- t = Tier
- 'U' = Unit
- 'S' = String
- 'C' = CAT
- 'T' = Training
- 'P' = Production
- 'E' = Exception (data base to hold all exception tables per system)
- DB = Literal
- nn = data base version number or designation
- 'II' = Original Production
- 'In' = Subsequent Production data base*
- nn = test version**

*NOTE

- The use of 'I1' should be avoided to help differentiate the database from the 'II' version.

**NOTE

- Test version numbers 01 through 19 are normally reserved for system versions, with numbers 20 through 99 and letters AA through ZZ assigned to individuals. Only databases (and their associated objects) which have been assigned system test version numbers will be migrated out of the unit testing tier. Individuals may use their individual databases (and associated objects) as they determine necessary for their development and testing requirements. However, neither objects nor data which exists in individual databases will be promoted to a higher tier of development.

- Test versions, which are intended to be used by batch processing or Client/Server applications only, will normally be assigned a version letter, instead of a number. This letter should correspond with the equivalent Tier and Chain of development.

EXAMPLE

- Development in the batch equivalent of the Chain A(1), String Tier would have a database name 'xxxSDB0A'.

- Development in the batch equivalent of the Chain B, String Tier would have a database name 'xxxSDB0C'.

- Test versions, which are intended to be used by both batch and online testing, will use the numeric version.

- Utility databases, developed and maintained in DB2 for use by Applications development (rather than the use by business applications) may use a less segmented naming convention. These databases will be identified and documented by the Data Base Management group as required.

TABLESPACES

Tablespaces will normally be defined as segmented and will normally contain one table. The tablespace name will be 8 bytes long and will use the following naming convention:

xxxTSnnn where:

- Xxx = Department (System) Acronym

- TS = Literal

- Nnn = Unique combination of letters and/or numbers. This combination will normally be a

number, and will be the next available number for that system.

TABLES

A table name may be any acceptable combination of letters and symbols for a DB2 table (up to 18 bytes). It may not, however, end with the characters '_S', '_V', or '_E'. Those trailing characters are reserved for synonyms, views and exception tables respectively (see the following items). The same name will be used in all test and production environments.

*NOTE

- Large agencies or applications should consider the use of a standard system prefix for each table. This prefix can help insure unique table names where the tables are not developed within a coordinated or central enterprise scheme.

INDEXES

An Index name may be any acceptable combination of letters and symbols for a DB2 index (up to 18 bytes). In general, a naming convention should be used which identifies the index as a primary and clustering index, a clustering and non-primary index, or a general support index.

EXAMPLE

xxx?tttn where:

- xxx = Department (System) Acronym
- ? = I for General support index,

  = P for primary and clustering index or primary index

  = C for clustering and non-primary index
- ttt = Unique combination of letters and/or numbers from tablespace name.
- n = a number or letter
- xxxI0051 or xxxI005A = the first general index for a table in tablespace '005'
- xxxP0052 or xxxP005B = the primary clustering index for a table in tablespace '005'
- xxxC0061 or xxxC006A = the clustering index for a table in tablespace '006' which is not a primary index.

SYNONYMS

Synonyms are unqualified pointers to physical tables. A synonym uses the current SQLID of the initiator to provide qualification. An SQL statement, which refers to a synonym uses the unqualified single node name of the synonym. DB2 then, uses the current SQLID and the synonym name as its entry into the catalog table SYSIBM.SYSSYNONYMS, and uses the resultant TBCREATOR and TBNAME columns for the actual physical table name.

Synonyms were once used exclusively in application programs because table names were actually changed between the development and production environments.

Where table names are not changed, there is no longer a need for this pointer capability to be used.

Synonyms should only be used when a table(s) from a different application system or with a different qualifier must be used. This situation typically occurs at the Unit or development level where the developer wants to mix tables, and where he/she uses his/her own SQLID as a qualifier.

The use of synonyms in Legacy systems will continue to be supported. The synonyms are normally limited to 8 bytes, with the fourth position of the name using the letter 'S' to represent the letters 'T' or 'P' in the actual table name.

Unless the synonym is used in this manner, and in a Legacy system, the use of synonyms is discouraged. Rather, the developer is encouraged to use actual table names, or in some cases, ALIASES or VIEWS in his/her SQL statements.

VIEWS

Views are generally used when a subset of table columns, or a group of columns from two or more tables (JOIN), are to be referenced in a single object. In normal practice, Views are necessary only when data security concerns must be considered and enforced without restriction, or when certain CASE tools require them to be able to construct code for JOIN operations. Views are limited to defining sets or subsets of tables which exist in the same subsystem in which the view is defined.

If used, a view will be any allowable name followed by the characters '_V' or the same name as the table but a V in the fourth position.

ALIASES

ALIASES are pointers to physical tables or VIEWS which can have a different qualifier and/or LOCATION than the table or VIEW to which it points. They are generally used in Client/Server applications where access authority must be specifically defined for individual groups of users for a limited group of tables. They are generally preferred in applications where dynamic SQL must be used in a developed application and where Distributed Relational Database Architecture (DRDA) protocols are utilized.

Unlike VIEWS, which require access GRANTS to be issued on the tables prior to the creation of the VIEW, access GRANTS on ALIASES may be made independently from the actual existence of the table to which it points. Further, an ALIAS can be created, and will exist, whether or not the table to which it points actually exists at the time of creation.

The authority to access data is GRANTED on the underlying table rather than on the ALIAS itself. This security consideration differs markedly from the VIEW which requires the creator of the VIEW to hold the authority on the underlying table, but who can pass on authority to access data from a table by GRANTING access to the VIEW, rather than granting access to the table itself.

If used, the Alias will use the same name as the table to which it points.

No other modifier or designation is required.

EXCEPTION TABLES

A Check Data Utility exception table will be the same name as the table for which it provides exception processing, followed by the characters '_E'. If the table name is over 16 bytes long, bytes 17 and 18 will be replaced with the characters '_E'. Or the same as the table name except an E in the fourth position.

DCLGEN

1. All DCLGEN's will be done on unqualified tables. Eight byte names will be assigned to the resulting Partitioned Data Set member for each of the tables. They will normally be the eight byte name used for the table's tablespace name, with the 'TS' replaced by the letters 'DG'. The 'G' portion of the name will be replaced by the letters 'B' through 'Z' in the event more than one table occupies the same tablespace. The actual name will be maintained in a Production cross reference DB2 table (DBMADM.CROSS_REFERENCE).

2. All objects which are referenced in SELECT, UPDATE, INSERT, and DELETE operations will be 'Declared' in the program.

3. Explicit column identification will be used for all SELECT, UPDATE, and INSERT operations.

OBJECT OWNERS, DATA OWNERS AND CREATORS

OBJECTIVE: Data Base Management will create all DB2 data bases, all DB2 objects within production system and development system data bases (version 00 - 19 numbered data bases in the development tiers), and will grant appropriate access to objects within those data bases to other selected ids.

1. RACF (or other authorized security package) will be used to create secondary authorization ids. It will also be used to control underlying data set access.

2. All DB2 tables, created by Data Base Management, will us a RACF secondary authorization group id as the 'Creator' of the tables. This secondary authorization group id will normally use an agency acronym for the first three positions of the id. For large projects, a project acronym may be used in place of the agency acronym. This full qualifier (as described below) will be used in the bind process to designate the qualifier of the tables referenced in the program or package. This qualifier will use the following naming convention:

xxxADtn where:

- xxx = Agency or Project Acronym
- AD = Literal
- t = Tier
- 'U' = Unit
- 'S' = String
- 'C' = CAT
- 'T' = Training
- 'M' = Production
- n = version or sequence letter or number *

* A version is only used in the Applications Development and Training environments.

Versions will not be used in the production environment.

EXAMPLE

DSSADS1 = Department of Social Services creator at the String Level, Chain A(1)

NSPADM = Nebraska State Patrol in the Production Level

3. Data Base Management, or the appropriate system RACF administrator, will establish secondary authorization ids for every system, subsystem, project, or major project cell.

4. Authorization id's will be created using the following naming convention:

Authorization Id = xxxuuts where:

- xxx = Department (System) Acronym
- uu = Type
- 'AD' = DBADM authority
- 'CL' = DBCTL authority
- 'AP' = Application
- 'US' = Client or User
- 'WA' = WEB Analyst
- 'WU' = WEB Client or User
- t = Tier
- 'U' = Unit
- 'S' = String
- 'C' = CAT
- 'T' = Training
- 'D' = Development
- 'P' = Production
- s = Test Path Designation Number*

*A number, or a letter, designating the testing 'Path' (e.g. '1', '2' or '3') will be used after the Tier designation. The number '0' will be used to designate access to all objects under the high level System acronym for application or task specific groups The letter 'T' will be used with the xxxADM designation to indicate project DBA authority in the entire test environment for that system.

In production this number will be used to separate differing levels of authorization.

Generally, a zero ('0') will indicate all authorities. A '1' or above will indicate specific authorities.

Individual TSO id's may be substituted for group id's for specific authorities.

*NOTE

- See Appendix A for an example of Owner and Qualifier names

5. The secondary authorization id groups will generally be assigned capabilities or accesses in the following areas:

- xxxADts = all DBADM authorities as outlined in the DB2 For OS/390 Administration Guide, Chapter 3. The DBM staff will normally be the only people placed into these groups.

- xxxCLts = DBCTL data base capabilities without the DROP, CREATETS, and CREATETAB authorities. People placed in these groups will normally be responsible for the maintenance of the DB2 tables in the development environment.

- xxxAPts = all access to table data including SELECT, UPDATE, INSERT, and DELETE. This group will generally contain the applications developers who are responsible for the development or maintenance of the programs within the system. These secondary authorization groups will generally have ownership of packages (i.e. programs) in the development environment.

- xxxUSts = SELECT authority only for the tables in a system. This group will generally contain the clients or users of the system who require ad-hoc READ access to the tables and data in the system.

- xxxWAts = EXECUTE authority on Web server plan. This group will generally contain the web analysts of the system to be able to call stored procedures through the WEB server.

- xxxWUts = EXECUTE authority on Web server plan. This group will generally contain the client users of the system to be able to call stored procedures through the WEB server.

6. Data Base Management will assign project staff to the secondary authorization groups in the production DB2 subsystems as required. Agency or system RACF administrators will assign project staff to the secondary authorization groups in the development subsystems for table authority and WEB server execute authority.

Project managers or their designates will be responsible for notifying Data Base Management when an individual is to be added or removed from a production subsystem secondary authorization group.

7. Data Base Management will provide master or sample DDL to create objects at the unit level. The project leader and Data Base Manager will jointly determine whether the project staff or Data Base Management will maintain objects at this unit level. If Data Base Management is selected to maintain objects at the unit level, it will only maintain objects in assigned system databases.

8. Data Base Management will be responsible for maintaining the DB2 objects in all test environments and Production environments. Exceptions to this provision may be approved by the Unit Supervisor to Data Base Management, when compelling reasons for those exceptions are warranted.

9. Data Base Management will do all DCLGEN's for production system and development system tables and will place the resultant copybooks or include members in the appropriate libraries.

10. Individuals will be responsible for all DB2 objects, which are defined, or developed in their personal version data bases. This responsibility includes table definitions and individual DCLGEN's.

11. Data Base Management will hold selected SYSCTRL authorizations, PACKADM, and DBADM control of all DB2 objects.

12. Except for normal maintenance capabilities possessed by the Data Base Management group, Client Agencies will maintain ownership over all data stored in DB2 objects and will have final approval regarding access to that data.

PLANS AND PACKAGES

1. Data Base Management will be responsible for creating all plans in both the test and production environments. Authorizations will then be granted to the appropriate Secondary Authorization ID.

2. In the Test DB2 environment, the following standards apply:

a) All plans will be created with an OWNER of DBMADM except WEB server plans. DBMADM then grants PACKADM on each collection to the appropriate level Authorization ID for the system, level and path. E.g. DBMADM grants PACKADM on collection RFSBATS1 to RFSAPS1. This authorization provides CREATE, BIND and EXECUTE for all packages in the collection using an owner of the Secondary Authorization ID for the system, level and path.

WEB server plans will be bound with DBMCLI. DBMCLI will grant EXECUTE on the WEB sever plan to the appropriate level Authorization ID for the system, level, and path. E.G. DBMCLI grants EXECUTE on plan CLIONLU1 to NFOWAU1.

b) The QUALIFIER of all plans and packages will be the Agency Secondary Authorization ID for the system, level and path. e.g. DSSADS1 would be the qualifier for the RFS plans in the String, Chain A(1) environment.

c) At the UNIT level, the QUALIFIER can be overridden to allow a program to access an individual's test tables. Example: using CCF/MMF, type 'DB2' on the command line of the CCF/MMF screen. This will display a DB2 override panel. Enter the owner of the tables you wish to access in the QUALIFIER field. NOTE: In order to do this, the following conditions must be met:

i. A table, alias or synonym for EVERY table accessed in the program must exist with a CREATOR equal to the QUALIFIER. E.g. if the program accesses TABLE_A and TABLE_B, and TABLE_B is owned by USERID, USERID must create a synonym called TABLE_A that points at the version of TABLE_A that is to be referenced.

ii. Access to the tables owned by the ID specified in the QUALIFIER must be granted to PUBLIC. E.g. if the program is going to per-form SELECTs and INSERTs on USERID.TABLE_A, then USERID must grant SELECT and INSERT on TABLE_A to PUBLIC.

3. In the Production DB2 Environment, the following standards apply:

a) The OWNER and QUALIFIER of all plans and packages in production will be the appropriate Agency Secondary Authorization ID for the system. e.g. DASADM is the owner and qualifier of the CCF production plans. NOTE: Before an Agency Secondary Authorization ID can bind a plan using Package Lists, it should be granted PACKADM on the system collections and EXECUTE on the DBM collections.

b) DBMADM grants EXECUTE on all online plans to PUBLIC except Web server plans, and EXECUTE on all batch plans to the appropriate Production ID used to execute scheduled jobs (e.g. CDPPROD, LABPROD, or DOR9999).

c) The Agency Secondary Authorization ID that owns the plan can optionally grant BIND AGENT to another Primary or Secondary Authorization ID. This authorization provides CREATE and BIND for all packages in collections on which the Agency Secondary Authorization ID holds PACKADM authority. e.g. DSSADM grants BIND AGENT to WEL7000, DASADM grants BIND AGENT to CCFBIND.

4. The plan name will normally be 8 bytes in length.

- The first three bytes will be the system acronym
- The second three bytes will be the environment
- The last two bytes of the plan name will correspond to the tier in which the plan is executed. The following naming convention will be used:

Plan name = xxxeeets where:

- xxx = Department (System) Acronym
- eee = Environmental designation
  - 'ONL' = Online
  - 'BAT' = Batch
- t = Tier designation
  - 'U' = Unit
  - S' = String
  - 'C' = Client Acceptance Test
  - 'T' = Training
  - 'A' = Assist GT
  - 'P' = Production
  - s = Test Path Designation Number (0 for production)

5. A program will be bound as a package and will normally be assigned to a collection-id, which has the same System Acronym. The mid-portion of the collection-id will be determined in the following manner:

a) An environment collection-id (e.g. XXXONLts, XXXBATts) when a module is only used in one environment (i.e. this name is normally the plan name under which the package is bound).

b) A combination batch and online collection-id (e.g. XXXCOMts) when a module is used by both the batch and online in the same system.

c) A Data Base Management collection-id (e.g. DBMBATts, DBMONLts or DBMCOMts) when a module is used in two or more systems, and where the module performs one or more universal functions which are not the specific function of a system process (e.g. date conversion routine, etc).

6. All Collection IDs will use the following naming convention:

Collection-id = xxxeeets where:

- xxx = Department (System) Acronym or 'DBM'
- eee = Environment (Online, Batch, or Combination)
- ONL = used only in Online programs
- BAT = used only in Batch programs
- COM = Combination Online and Batch
- t = Tier designation
- 'U' = Unit
- 'S' = String
- 'C' = Client Acceptance Test
- 'T' = Training
- 'A' = Assist GT
- 'P' = Production
- s = Test Path Designation Number (0 for production)

OBJECT CREATION AND AUTHORIZATIONS

When creating objects and granting authorizations, the following objectives should be kept in mind.

- Objects should not be created with an individual ID as the CREATOR or OWNER. The CREATOR should be a RACF group to which the DBM staff is as- signed. The recommended CREATOR of each object type will be identified in the following sections.

- Grants should not be made from individual IDs, but instead should be made from a RACF group. Those rare grants that require SYSADM authority should be made from the DBMZ001 ID. This will allow any of these special grants and revokes to be controlled by the DBM group.

- Wherever possible, grants should be made with the CURRENT SQLID set to DBMADM. This should minimize the need to constantly change the CURRENT SQLID. Because some objects are not owned by DBMADM, there are cases where DBMADM does not have the authority to perform the grant. In these cases, the authority should be granted from the owner to DBMADM with grant, and then subsequent authorizations should be granted from DBMADM. The instances where this is necessary will be identified in the following sections.

- The grants identified in this document are the minimum, or default, authorization levels. Additional grants may be made, but generally only with an approved DB2 RACF request. Where appropriate, these optional authorizations will also be documented.

- Exceptional or temporary grants should be made to individual IDs. Examples of this include the database authorizations that are permanently required by CDPPROD to run scheduled utilities, and temporary authorities such as LOAD on a production database or EXECUTE on a production batch plan.

- SYSADM authority should be used to grant only where documented. The only grants that have been identified that must be done with SYSADM authority are the initial system authorizations that are granted to DBMADM. If additional authorizations must be granted using SYSADM authority, they should be documented here. This point is not in reference to the RACF group named SYSADM.

- The old RACF group that is named SYSADM should not be used to perform any new grants. This RACF group generates confusion since it is named the same as a DB2 authorization level that it no longer holds. There are still a number of objects that have SYSADM as the creator. In these cases, authority on the object should be granted from SYSADM to DBMADM with grant, and then subsequent grants can be made from DBMADM.

- Whenever possible, while objects are being maintained or migrated, if the creators of objects or grantor of authorizations are different than the recommendations documented here, replace them.

- On the surface, these recommendations may appear to be merely replacing SYSADM as the owner of objects and grantor of authorizations with DBMADM. There are several reasons for the change that explain why this is not the case.

- Previously, SYSADM was the owner of objects and grantor of most authorizations since this RACF group held SYSADM authority. There is no need to use SYSADM authority on a daily basis if the proper grants are made. In fact, it could be rather dangerous to constantly have our SQLID's set to a RACF group that has SYSADM authority. The reason DBMADM has been chosen as the source of most of the grants is to prevent the confusion of having a RACF group named the same as a DB2 system authorization. At some point in the future, the SYSADM RACF group can probably be eliminated.

- Having SYSADM as the creator of all of the tables meant that a given table name could only be used once in the entire system. Because of the number of agencies that we support, and their independence from each other, it is preferable to treat each agency as a separate enterprise. By using an agency level qualifier, agencies no longer need to worry about colliding with a table name used by another agency.

- Some agencies have the need to perform their own production binds. Previously, this would require that these agencies hold the authorities to create and own the plans, collections and packages. In many cases, we would then not have sufficient authority to rebind their plans and packages when performing maintenance (without using SYSADM). By using the agency level owners and bind agent authority, we are able to allow these agencies to run their own binds and still maintain control of the plans, collections and packages.

See Appendix C for a matrix of the authorizations listed and described below.

SYSTEM AUTHORIZATIONS

There should be two IDs identified as Installation SYSADM on each subsystem. Currently, these IDs are DBMZ001 and Arlan Walvoord's ID, which is CDP0013. The DBMZ001 ID should be used to grant all necessary authorities to DBMADM, and then subsequent grants should originate from there.

The following authorities should be granted from DBMZ001 to DBMADM on all subsystems:

- SYSCTRL WITH GRANT OPTION

- PACKADM ON COLLECTION * WITH GRANT OPTION

The following SYSTEM authorities should be granted from DBMADM on all subsystems:

- DISPLAY, MONITOR1, MONITOR2, TRACE TO PUBLIC, PTXMAN

On the DBT subsystem, the following grant should also be made from DBMADM to allow Dept. of Roads to create their own databases:

- CREATEDBA TO DORDBAM

On the DBT and DB2 subsystems, BIND AGENT authority is used to allow the binds of plans and packages without changes in ownership.

On the DBT subsystem, each aaaADtp should grant BIND AGENT TO PUBLIC to authorize binds at all test levels.

On the DB2 subsystem, each aaaADM should grant BIND AGENT TO ZQGTCDX and to the aaaBIND RACF group.

BUFFERPOOL AUTHORIZATION

The SYSCTRL authority held by DBMADM includes USE on all defined bufferpools. Therefore, as long as DBMADM authority is used when creating tablespaces and indexes, no additional grants are required for us to maintain these objects in any bufferpool. However, since individuals can create these objects under certain circumstances, USE OF BUFFERPOOL BP0 should be granted from DBMADM to PUBLIC on all subsystems.

DBT SUBSYSTEM BUFFERPOOLS

Since the Charts and N-Focus DBAs are maintaining their own objects in bufferpools BP3 and BP4 on DBT, USE of these two bufferpools should be granted to CSECLU1, CSECLU4 and NFOADMT.

STOGROUP CREATION AND AUTHORIZATION

All STOGROUPs on all subsystems should be created by DBMADM. As the CREATOR of the STOGROUP, DBMADM implicitly holds all authority on that STOGROUP with grant ability. For old STOGROUPs that have a CREATOR other than DBMADM, the STOGROUP creator should grant USE on the STOGROUP to DBMADM WITH GRANT OPTION. Subsequent grants can then be made from DBMADM.

DBT SUBSYSTEM STOGROUPS

On DBT, USE ON STOGROUP SYSDEFLT should be granted to PUBLIC, and USE ON STOGROUP DSSSTOG1 should be granted to CSECLU1, CSECLU4 and NFOADMT.

DATABASE CREATION AND AUTHORIZATION

With the exception of the databases created by Dept. of Roads on DBT, all databases should be created by DBMADM. As the CREATOR of the database, DBMADM implicitly holds all authority on that database with grant ability. For legacy databases that have a creator other than DBMADM, the database creator should grant DBADM TO DBMADM WITH GRANT OPTION. Subsequent grants can then be made from DBMADM.

- INDIVIDUAL DATABASES - (sssUDBxx) where xx is the individual's initials.

When databases are created for individuals to use, DBMADM should grant DBADM WITH GRANT to the individual's MVS ID. This will allow the individual to perform all operations within the database, and grant authority to whoever else may need it.

- UNIT LEVEL DATABASES - (sssUDB0p)

For unit level databases, DBMADM should grant DISPLAYDB, IMAGCOPY, LOAD, RECOVERDB, REORG, REPAIR, STARTDB, STOPDB, STATS to the CL RACF group for that system, tier and path (sssCLtp).

- STRING AND CAT LEVEL DATABASES - (ssstDB0p)

At the string and CAT levels, DBMADM should grant DISPLAYDB, IMAGCOPY, LOAD, RECOVERDB, REORG, REPAIR, STARTDB, STOPDB, STATS, TO CDPPROD and to the CL RACF group for that system, tier and path. (sssCLtp)

- TRAIN AND ASSIST LEVEL DATABASES - (ssstDB0p)

At the Training and Assist/GT levels, DBMADM should grant DISPLAYDB, IMAGCOPY, LOAD, RECOVERDB, REORG, REPAIR, STARTDB, STOPDB, STATS, TO CDPPROD and to the AWPCL group for the tier and path. (AWPCLtp)

- PRODUCTION LEVEL DATABASES - (sssPDBIx)

At the production level, DBMADM should grant DISPLAYDB, IMAGCOPY, REORG, REPAIR, STARTDB, STOPDB and STATS to CDPPROD. Additional authorities on production databases can be requested via the DB2RACF form, and should be granted from DBMADM to the individual's IDs.

TABLESPACE CREATION AND AUTHORIZATION

All tablespaces controlled by the DBM group should be created by DBMADM. No further authorizations are required.

TABLE CREATION AND AUTHORIZATION

All tables should be created using DBMADM authority, but with an agency level qualifier. As the creator of the database, DBMADM holds the necessary CREATE table authority. Using an agency level qualifier allows each agency to operate as a separate enterprise and name their tables without regard to table names used in other agencies. In production, an agency qualifier in the form aaaADM should be used. The agency level qualifier of all test tables should also include the tier and path designation: aaaADtp.

- UNIT LEVEL TABLES - (aaaADUp.table_name)

At the unit level, DBMADM should grant SELECT, INSERT, DELETE and UPDATE on all tables to it’s associated RACF group, xxxAPtp.

- STRING, CAT, TRAIN and ASSIST/GT LEVEL TABLES - (aaaADtp.table_name)

At each of these levels, DBMADM should grant SELECT, INSERT, DELETE and UPDATE on each database's tables to it's associated applications RACFgroup, sssAPtp.

- PRODUCTION LEVEL TABLES - (aaaADM.table_name)

At the production level, DBMADM should grant SELECT, INSERT, DELETE and UPDATE on each database's tables to it's associated sssAPP0 RACF group. DBMADM should also grant SELECT to the sssAPP1 and sssUSP0 RACF groups. Users requiring "standard" access to the production tables will be placed into one of these existing RACF groups upon receipt of an approved DB2RACF request form.

sssAPP0 - Applications personnel requiring update capability.

sssAPP1 - Applications personnel requiring read-only access.

sssUSP0 - End-users requiring read-only access.

Users requiring "non-standard" access to the production tables (i.e. end-user update or access to selected tables only) may be placed in a special RACF group or may have the authority granted directly to their ID. Any of these grants should originate from the DBMADM RACF group.

INDEX CREATION AND AUTHORIZATION

Indexes should be created in the same way as tables, using DBMADM authority and agency level qualifiers. No additional authorizations are required.

ALIAS CREATION AND AUTHORIZATION

When used, Aliases should be created using DBMADM authority and an agency level qualifier. No additional authorizations are required.

VIEW CREATION AND AUTHORIZATION

Views should be created in the same was as tables, using DBMADM authority and agency level qualifiers. Immediately after creation, only the creator of a view holds authority on it. Therefore, the agency level RACF group (aaaADM or aaaADtp) must grant SELECT on the view to DBMADM with grant option. Then, all subsequent grants can be made using DBMADM authority.

- UNIT LEVEL VIEWS - (aaaADUp.view_name)

At the unit level, DBMADM should grant SELECT on all views to it's associated applications RACF

Group, sssAPtp.

- STRING, CAT, TRAIN and ASSIST/GT LEVEL VIEWS - (aaaADtp.view_name)

At each of these levels, DBMADM should grant SELECT on each database's views to it's associated applications RACF group, sssAPtp.

- PRODUCTION LEVEL VIEWS - (aaaADM.table_name)

At the production level, DBMADM should grant SELECT on each database's views to it's associated sssAPP0, sssAPP1 and sssUSP0 RACF groups. See the PRODUCTION LEVEL TABLES section for an explanation of these RACF groups, and rules for additional authorizations.

COLLECTION AUTHORIZATION

Collections are not explicitly created, they exist as a logical foreign key on the SYSPACKLIST and SYSPACKAGE catalog tables. Before a plan or package can be bound using the collection, however, the appropriate authorizations must be in place. Even with SYSCTRL authority, DBMADM does not have the authority to bind a plan using a package list, or bind a package. DBMZ001 must first grant DBMADM PACKADM authority on all collections (*) with grant option. In conjunction with the BIND ADD authority that is part of SYSCTRL, this will enable DBMADM to bind plans using package lists.

- UNIT STRING, CAT, TRAIN AND ASSIST/GT LEVEL COLLECTIONS - (sssONLtp, sssBATtp)

At each testing level, DBMADM should grant PACKADM on each collection to it's associated agency RACF group (aaaADtp). In conjunction with the grant of BIND AGENT from each of the agency RACF groups to PUBLIC, this will allow anyone to bind packages at all test levels.

- PRODUCTION LEVEL COLLECTIONS - (sssONLP0/sssBATP0)

At this level, DBMADM should grant PACKADM on each collection to it's associated production agency RACF group (aaaADM). In conjunction with and grants of BIND AGENT done from each agency RACF group, this will provide the ability to bind production packages, but only within the agency for which BIND AGENT was granted.

PLAN CREATION AND AUTHORIZATION

With the exception of applications that use dynamic plan allocation, all plans on all subsystems should be created with an OWNER of DBMADM except WEB server plans. This allows us to grant and revoke all plan authorities from this RACF group. Previously, when some plans were owned by the agency level group, users who had been granted BIND AGENT authority were able to rebind the plans with different parameters, and there is not a need for this ability.

- TEST PLANS - (sssONLtp/sssBATtp)

EXECUTE on all test (UNIT, STRING, CAT, TRAIN & ASSIST) plans should be granted from DBMADM to PUBLIC.

- PRODUCTION BATCH PLANS - (sssBATP0)

EXECUTE on all production batch plans should be granted from DBMADM to CDPPROD. This will allow all production plans to be executed as part of a scheduled batch job. EXECUTE authority on production batch plans can be requested via the DB2RACF form, and should be granted from DBMADM to the individual ID. This is generally a temporary authorization that should be revoked after a pre-determined number of days.

- PRODUCTION ONLINE PLANS - (sssONLP0)

EXECUTE on all production online plans should be granted from DBMADM to PUBLIC, since we will rely on CICS security to prevent unauthorized access. Because of this, all production online plans should ENABLE (CICS) to prevent access from any other attachment.

- PRODUCTION ONLINE PLANS - DYNAMIC PLAN ALLOCATION

Plans that are used via dynamic plan allocation are still DBRM binds and need to be bound with an aaaADM owner. This will enable those individuals granted BIND AGENT from the aaaADM to bind these plans. The aaaADM RACF group that owns these plans should grant EXECUTE on the plans with grant option to DBMADM. Then, the grant of EXECUTE to PUBLIC can be made from DBMADM.

- TEST and PRODUCTION WEB SERVER PLANS - (CLIONLtp)

EXECUTE on Web server plan will be granted granted from DBMCLI to sssWAtp and sssWUtp. This will allow only analysts and users connected to these RACF groups to be able to call stored procedures through the WEB server.

PACKAGE CREATION AND AUTHORIZATION -

(sssONLtp/sssBATtp.package)

There are no additional authorizations required at the package level. The authority to bind and execute packages is controlled at the collection and, indirectly, at the plan level. This section will address what OWNER and QUALIFIER should be used for packages at each level.

- UNT, STG, CAT, TRAIN AND ASSIST/GT LEVEL PACKAGES (sssONLtp/sssBATtppkg)

Packages at all testing levels should be bound with and agency level OWNER and QUALIFIER that includes the tier and path (aaaADtp).

- PRODUCTION PACKAGES - (sssONLP0/sssBATP0.package)

Production packages should be bound with an agency level OWNER and QUALIFIER (aaaADM).

AUTHORIZATIONS REQUIRED FOR OBJECT PROMOTION

OBJECTIVE: Data Base Management will promote database objects, plans and packages only with the approval of the appropriate project person.

1. DB2 objects, plans and packages will be promoted to the next level of test or production only with the signature of the responsible person as listed below:

a) String = Project Analyst

b) CAT = Lead Analyst

c) Training = Lead Analyst

d) Production = Project Manager

2. The next higher authority may sign off objects for a lower authority. A lower authority can only be assigned to provide signatures if approved by a higher authority than the person for whom the provision is being made (e.g. The Project Manager may approve a Project Analyst signing plans into CAT in the absence of the Lead Analyst).

3. Production plan or package binds may be accomplished by an Oncall analyst or programmer, but only to replace a plan or package currently in production. These actions require the signature of the Project Manager by the close of the next business day. In these instances, the signature insures that the manager is aware of the emergency and the actions taken to remedy the production problem. Failure to secure this or another approved signature will result in the plan or package being freed.

PRIORITY AND TIMELINESS OF OBJECT PROMOTIONS

OBJECTIVE: Data Base Management will promote data base objects, plans and packages in a timely manner.

1. Proper and adequate planning contributes to the successful implementation of database objects or programs. To this end, sufficient prior notification of schedules to move objects or programs must be given.

2. The following matrix provides a reasonable time-frame for the movement of objects through the test levels:

a) Plan or Package bind = 1 hour per program

b) Table alter = 1 hour per object per alter

c) Table or Index create = 1 hour per object

3. The following matrix provides a reasonable time-frame for the movement of objects to production after notification:

a) Plan or Package bind = 1 hour per program

b) Table alter = 1 hour per object

c) Table or Index create = 1 hour per object

4. These changes can be expected to be made at the rate of 1 movement per hour. The project leader should tally all movements and compute the lead-time necessary to move these objects through test and production. In those cases where other system changes must be made, or where changes must await system disconnects and reloads, these changes or moves may be delayed until the next cycle unit.

5. Unless written notice and explanation is provided, all Table additions or changes will be accomplished WITHOUT data recovery. Sufficient notice and analyst cooperation will be needed to accomplish changes with data recovery.

USE OF AUTHORIZED TOOLS FOR OBJECT PROMOTION

OBJECTIVE: Data Base Management will use only authorized software tools or products to promote database objects.

1. Promotions or the movement of objects through the test and production tiers will be accomplished using only authorized Data Base Management software tools or products (e.g. Platinum, ADW, etc).

USE OF AUTHORIZED SOFTWARE FOR PROGRAM PROMOTIONS AND BINDS

OBJECTIVE: A common methodology and software will be used to promote programs and accomplish DB2 binds, at all levels of development and into production.

1. Promotions or the movement of objects through the test and production tiers will be accomplished using the CCF/MMF Migration methodology and software. See Appendix C for a description of the promotion strategies to be used for normal and emergency migrations.

TRANSITION

OBJECTIVE: A Data Base Management staff person will ensure that all database objects, and appropriate copy and recovery procedures are in a receiving environment during the transition and promotion phases of database objects.

1. Data Base Management will be responsible for all maintenance and recovery procedures for DB2 objects in the production environment (see Document 93008CDP0194, DB2 Data Base Management Systems Procedures).

2. The CCF staff will be responsible for the migration of normal Application Development programs and associated objects through the CCF chains.

3. All conversion and/or one-time programs must be successfully executed in a String environment. Data Base Management must approve the results of these programs before they may be promoted and run in the production environment.

4. The DB2 Load Utility will be used for large or mass inserts of information into DB2 tables. Those jobs will use the 'LOG NO' parameter (i.e. do not log the inserts) for those inserts. Data Base Management may authorize exceptions to this procedure when processing requirements justify that exception.

5. All Migrations and Conversions of DB2 objects will use a formal and written script to plan, implement, and verify the movement or change of DB2 objects in the production environment. Appendix D, DB2 Migration and Conversion Procedures Checklist, may be used as a skeleton script. Alterations, revisions, additions, and omissions should be made based upon the unique requirements of each project.

PRODUCTION

OBJECTIVE: Data Base Management will ensure that all access to data base objects, and information contained in those objects is done only for the business of the State, is done in a manner which ensures the integrity of the data base objects, and the data contained in those objects, and is accessed in the most efficient and secure manner possible.

1. Data Base Management will direct or accomplish timely maintenance of DB2 tablespaces and their indexes. This maintenance includes:

- Regular image copies of all tablespaces.
- Resizing of tablespaces or index spaces as required.
- Reorganization of tablespaces to effect changes to those data sets.

2. DB2 objects, including tablespaces, tables, and indexes will only be recovered using image copies and/or system logs when those objects have been rendered unusable by a System or DASD device failure, and when no other means to recover the object is available. These extraordinary recovery procedures will not be used to reverse the alteration of data within those objects, unless that recovery is directed and approved by director(s) of the responsible business areas and the senior manager(s) of CDP.

CHANGING PRODUCTION TABLES AND PROGRAMS

OBJECTIVE: Production tables will normally be changed using the DB2 ALTER option rather than using the DROP option.

1. A table which is defined in production, and which contains viable business data, will normally not be deleted (DROPPED) and redefined to effect changes to column name or order. In situations where it is felt that changes in column names or orders must be made, the creation and use of a VIEW is preferred.

2. In situations where changes to column names and column order cannot be accommodated with the use of a VIEW, or when the data type of a column must be changed and the table must be DROPPED and CREATED, the following responsibilities will be observed:

- The responsible application development area will schedule an outage with all effected parties, including all user, technical support, and operations staff.

- The responsible application development area will unload the table data using either the standard DSNTIAUL unload program, or an equivalent program.

- DBM will DROP the old table and CREATE the new table based upon the information furnished by the responsible application area.

- The responsible application development area will convert the old data to the new format.

- The responsible application development area will load the converted data in its new format.

- DBM will arrange for standard Image Copy and, if necessary, CHECK DATA utilities to be run.

- The responsible application development area will examine the output from the CHECK DATA utility and resolve all rejected data or consistency conflicts.

- The responsible application development area will examine the data for accuracy and consistency.

- DBM will arrange for standard RUNSTAT and BIND utilities to be run.

- DBM will remove all prior references to Image Copy data sets from the SYSIBM.SYSCOPY table which were made prior to the DROP table action.

- The responsible application development area will insure that the new table, the effected programs, QMF queries, and procedures are operational.

3. In situations where changes to tables and programs will be made through the DROP and CREATE of a table, DBM will refer to the checklists found in Appendix D, "DB2 MIGRATION/CONVERSION CHECKLIST" to guide and suggest the actions which should be followed in the conversion process.

POST IMPLEMENTATION REVIEW

OBJECTIVE: Data Base Management will conduct a Post Implementation Review for all new systems or for all projects which significantly changed a current system.

1. A Post implementation review will be held three (3) to six (6) months after a project has gone into production.

2. Participants in this review will include staff involved in the design, construction, implementation, and operation of the new or changed system.

3. This review will cover, but not be limited to, the following topics:

- Recommended changes to Standards, Procedures, and/or Guidelines
- System strengths and deficiencies
- Development software and tool strengths and deficiencies
- Plans to correct design and/or performance problems

APPENDIX A

APPL-ID (OWNER)	DATA BASE NAME	TABLESPACE NAME	QUALIFIER	TABLE NAME	
UNIT	DSSADU1	MCPUDB01	MCPTS001	DSSADU1	anyname
STRING	DSSADS1	MCPSDB01	MCPTS001	DSSADS1	anyname
CAT	DSSADC1	MCPCDB01	MCPTS001	DSSADC1	anyname
PROD	DSSADM	MCPPDBII	MCPTS001	DSSADM	anyname

- The preceding example uses the Medicaid system, which has a system acronym 'MCP' and an Agency acronym of 'DSS'.
- The Owner of all packages bound at all test environments would be DSSADts as executed by the CCF group using their BIND AGENT authority.
- The Qualifier for all test tables/indexes/views would be 'DSSADts'.
- The Qualifier for all Production tables would be 'DSSADM'.

CICS REGIONS and DB2

APPENDIX B

OTHER SYSTEMS DEVELOPMENT CHAINS

PATH 1 **NOT USED** **PATH 3**

UNIT ESAUNT1 MVSUNT1 ESAUNT3 MVSUNT3

TESTCIC1 TESTMCI1 TESTCICD TESTMCID

DSNCRCTA DSNCRCTA DSNCRCTM DSNCRCTM

CHAIN A **NOT USED** **CHAIN B**

DB = sssUDB01 DB = sssUDB03

QUAL= aaaADU1 QUAL= aaaADU3

GRP = sssAPU1 GRP = sssAPU3

PLAN= sssONLU1 PLAN= sssONLU3

STRING ESASTG1 MVSSTG1 ESASTG3 MVSSTG3

TESTCIC2 TESTMCI2 TESTCIC9 TESTMCI9

DSNCRCTB DSNCRCTB DSNCRCTI DSNCRCTI

CHAIN A **NOT USED** **CHAIN B**

DB = sssSDB01 DB = sssSDB03

QUAL= aaaADS1 QUAL= aaaADS3

GRP = sssAPS1 GRP = sssAPS3

PLAN= sssONLS1 PLAN= sssONLS3

CAT ESACAT1 MVSCAT1 ESACAT3 MVSCAT3

TESTCIC3 TESTMCI3 TESTCICA TESTMCIA

DSNCRCTC DSNCRCTC DSNCRCTJ DSNCRCTJ

CHAIN A **NOT USED** **CHAIN B**

DB = sssCDB01 DB = sssCDB03

QUAL= aaaADC1 QUAL= aaaADC3

GRP = sssAPC1 GRP = sssAPC3

PLAN= sssONLC1 PLAN= sssONLC3

CURRENT PRODUCTION **INCLUDES CAT CHANGES**

TRAIN ESATRN1 MVSTRN1 ESATRN3 MVSTRN3

TESTCIC4 TESTMCI4 TESTCIC8 TESTMCI8

DSNCRCTD DSNCRCTD DSNCRCTH DSNCRCTH

CHAIN A **CHAIN B**

DB = sssTDB01 DB = sssTDB03

QUAL= aaaADT1 QUAL= aaaADT3

GRP = sssAPT1 GRP = sssAPT3

PLAN= sssONLT1 PLAN= sssONLT3

INCLUDES CAT CHANGES

ASSIST GT ESAAST1 MVSAST1 ESAAST3 MVSAST3

TESTCIC5 TESTMCI5 TESTCIC? TESTMCI?

DSNCRCTE DSNCRCTE DSNCRCT? DSNCRCT?

CHAIN A **CHAIN B**

DB = sssADB01 DB = sssADB03

QUAL= aaaADA1 QUAL= aaaADA3

GRP = sssAPA1 GRP = sssAPA3

PLAN= sssONLA1 PLAN= sssONLA3

sss = System Acronym (e.g. MCP, PAE, NAS, etc.)

aaa = Department Acronym (e.g. DSS, NSP, DAS, etc.)

ESA RCT = CICS.TEL.SOURCE MVS RCT = CICS.ICSOURCE

CICS REGIONS and DB2

APPENDIX B

N-FOCUS DEVELOPMENT CHAINS

PATH 4 **PATH 2** **NOT USED**

UNIT ESAUNT4 MVSUNT4 ESAUNT2 MVSUNT2

TESTCICE TESTMCIE TESTCICB TESTMCIB

DSNCRCTN DSNCRCTN DSNCRCTK DSNCRCTK

CHAIN C **CHAIN D** **NOT USED**

DB = sssUDB04 DB = sssUDB02

QUAL= aaaADU4 QUAL= aaaADU2

GRP = sssAPU4 GRP = sssAPU2

PLAN= sssONLU4 PLAN= sssONLU2

STRING ESASTG4 MVSSTG4 ESASTG2 MVSSTG2

TESTCICF TESTMCIF TESTCIC6 TESTMCI6

DSNCRCTO DSNCRCTO DSNCRCTF DSNCRCTF

CHAIN C **CHAIN D** **NOT USED**

DB = sssSDB04 DB = sssSDB02

QUAL= aaaADS4 QUAL= aaaADS2

GRP = sssAPS4 GRP = sssAPS2

PLAN= sssONLS4 PLAN= sssONLS2

CAT ESACAT4 MVSCAT4 ESACAT2 MVSCAT2

TESTCICG TESTMCIG TESTCIC7 TESTMCI7

DSNCRCTP DSNCRCTP DSNCRCTG DSNCRCTG

CHAIN C **CHAIN D** **NOT USED**

DB = sssCDB04 DB = sssCDB02

QUAL= aaaADC4 QUAL= aaaADC2

GRP = sssAPC4 GRP = sssAPC2

PLAN= sssONLC4 PLAN= sssONLC2

sss = System Acronym (e.g. MCP, PAE, NAS, etc.)

aaa = Department Acronym (e.g. DSS, NSP, DAS, etc.)

ESA RCT = CICS.TEL.SOURCE MVS RCT = CICS.ICSOURCE

AUTHORITY MATRIX

APPENDIX C

SYSTEM AUTHORIZATIONS

DBX	DBT	DB2	DB3
INSTALL SYSADM	DBMZ092	DBMZ092	DBMZ092	DBMZ092
DBMZ001	DBMZ001	DBMZ001	DBMZ001
INSTALL SYSOPR			
ARCHIVE			
BINDADD			
BINDAGENT	aaaADTp>PUBLIC	aaaADM>aaaBIND	
aaaADM>ZQGTCDX			
BSDS			
CREATEALIAS			
CREATEDBA	DBMADM>DORDBAM		
CREATEDBC			
CREATESG			
DISPLAY			
MONITOR1	PUBLIC PUBLIC		
MONITOR2	PTXMAN	PTXMAN	PTXMAN	PTXMAN
PUBLIC PUBLIC PUBLIC			
RECOVER			
STOPALL			
STOSPACE			
SYSADM			
SYSCTRL	DBMADM	DBMADM	DBMADM	DBMADM
SYSOPR			
TRACE	PTXMAN	PTXMAN	PTXMAN	PTXMAN
PUBLIC PUBLIC PUBLIC			

BUFFERPOOLS

USE OF BP0 PUBLIC PUBLIC PUBLIC PUBLIC

USE OF BP3,BP4 CSECLU1/4

NFOCLU4

STOGROUPS

CREATOR DBMADM DBMADM DBMADM DBMADM

USE OF SYSDEFLT PUBLIC

USE OF DSSSTOG1 CSECLU1/4

NFOCLU4

AUTHORITY MATRIX

APPENDIX C

UNIT STRING/CAT TRAIN/ASSIST PRODUCTION

DATABASES

CREATOR DBMADM DBMADM DBMADM DBMADM

DBADM

DBCTRL

DBMAINT

CREATETAB

CREATETS

DISPLAYDB PUBLIC PUBLIC PUBLIC PUBLIC

DROP

IMAGCOPY PUBLIC sssCLTp sssCLTp CDPPROD

CDPPROD CDPPROD

LOAD PUBLIC sssCLTp sssCLTp CDPPROD

CDPPROD CDPPROD

RECOVERDB PUBLIC sssCLTp sssCLTp CDPPROD

CDPPROD CDPPROD DBMADM

REORG PUBLIC sssCLTp sssCLTp CDPPROD

CDPPROD CDPPROD DBMADM

REPAIR PUBLIC sssCLTp sssCLTp CDPPROD

CDPPROD CDPPROD DBMADM

STARTDB PUBLIC sssCLTp sssCLTp CDPPROD

CDPPROD CDPPROD DBMADM

STOPDB PUBLIC sssCLTp sssCLTp CDPPROD

CDPPROD CDPPROD DBMADM

STATS PUBLIC sssCLTp sssCLTp CDPPROD

CDPPROD CDPPROD DBMADM

TABLESPACES

CREATOR DBMADM DBMADM DBMADM DBMADM

USE OF

TABLES

CREATOR aaaADtp aaaADtp aaaADtp aaaADM

ALTER

DELETE PUBLIC sssAPtp sssAPtp sssAPP0

INDEX

INSERT PUBLIC sssAPtp sssAPtp sssAPP0

SELECT PUBLIC sssAPtp sssAPtp sssAPP0

sssAPP1

sssUSP0

UPDATE PUBLIC sssAPtp sssAPtp sssAPP0

AUTHORITY MATRIX

APPENDIX C

VIEWS

CREATOR aaaADtp aaaADtp aaaADtp aaaADM

SELECT PUBLIC sssAPtp sssAPtp sssAPP0

SssAPtp sssAPP1

sssUSP0

DBMADM W/G DBMADM W/G DBMADM W/G DBMADM W/G[3]

INSERT

UPDATE

DELETE

ALIASES

CREATOR aaaADtp aaaADtp aaaADtp aaaADM

COLLECTIONS

CREATE

PACKADM aaaADtp aaaADtp aaaADtp aaaADM

ONLINE PLANS

CREATOR DBMADM DBMADM DBMADM DBMADM

QUALIFIER aaaADtp aaaADtp aaaADtp aaaADM

BIND

EXECUTE PUBLIC PUBLIC PUBLIC PUBLIC

BATCH PLANS

CREATOR DBMADM DBMADM DBMADM DBMADM

QUALIFIER aaaADtp aaaADtp aaaADtp aaaADM

BIND

EXECUTE PUBLIC PUBLIC PUBLIC CDPPROD

WEB PLANS

CREATOR	DBMCLI	DBMCLI	DBMCLI	DBMCLI
QUALIFIER	DBMCLI	DBMCLI	DBMCLI	DBMCLI
BIND				
EXECUTE	aaaWAtp	aaaWAtp	aaaWAtp	aaaWAP0
	aaaWUtp	aaaWUtp	aaaWUtp	aaaWUP0

PACKAGES

CREATOR	DBMADM	DBMADM	DBMADM	DBMADM
QUALIFIER	aaaADtp	aaaADtp	aaaADtp	aaaADM
ALL				
BIND				
COPY				
EXECUTE				
RUN				

CCF/MMF MIGRATION

APPENDIX D

NORMAL MIGRATION

When a request is made to move a program from CAT to STA(ge), the following actions must take place:

1. The SOURCE must be moved to STA.SRCMAST.LIBR from one of development libraries:

- CAT.CHAINx.SRCMAST.LIBR

2. The LOAD MODULE must be moved from:

- CAT.CHAINx.BAT.LOADLIB to TPD.BAT.LOADLIB

or

- CAT.CHAINx.CICS.ESA.LOADLIB to TPD.CICS.ESA.LOADLIB

or

- CAT.CHAINx.CICS.MVS.LOADLIB to TPD.CICS.MVS.LOADLIB

3. The Data Base Request Module (DBRM) must be moved to TPD.DBRMLIB from the DBRM library:

- CAT.CHAINx.DBRMLIB

4. The SOURCE which was moved to STA.SRCMAST.LIBR must be compiled creating both of the following:

- A DBRM in STA.DBRMLIB
- A Batch LOAD in STA.BAT.LOADLIB

or

- An Online LOAD in STA.CICS.MVS.LOADLIB or STA.CICS.ESA.LOADLIB.

5. The PRD DBRM and LOAD must be copied from PRD to PRE (first).

6. The STA DBRM and LOAD must be copied from STA to PRD (second).

7. A Package Bind must be done in production using the STA DBRM.

8. A Package Bind must be done in every test STR(ing) and CAT region, and in all training regions, using the concatenation of the CHAIN and TIER DBRM library, of that region and above, and the TPD DBRM library.

CCF/MMF MIGRATION

APPENDIX D

NORMAL MIGRATION OF SOURCE AND MODULES

TEST

CAT	DBRM	SOURCE	LOAD
TPD	DBRM		LOAD
MOVED		MOVED	

PRODUCTION

STA	DBRM	MOVED	LOAD
(1)	CREATED	COMPILED	CREATED
PRD	DBRM		LOAD
COPIED		COPIED	
(3)	from STA 2nd		from STA 2nd
PRD	DBRM		LOAD
COPIED		COPIED	
(2)	from PRD 1st		from PRD 1st

CCF/MMF MIGRATION

APPENDIX D

SUMMARY of NORMAL MIGRATION

1. The LOAD and DBRM in CAT are moved to TPD and become the basis for all executions and binds for all levels in the Test and Training environments.

2. Only the SOURCE is moved to STAGE where the precompile and the link create new LOAD and DBRM which are the basis for all executions and binds in the Production environment. The precompile uses the 'VERSION (AUTO)' feature. ALL PRODUCTION MODULES ARE CREATED FROM ACTIONS WHICH START WITH THIS SOURCE.

3. The current Production (PRD) LOAD and DBRM are moved to the Previous (PRE) libraries.

4. The new Stage (STA) LOAD and DBRM are copied to the current Production (PRD) libraries.

CCF/MMF MIGRATION

APPENDIX D

EMERGENCY MIGRATION

An Emergency Migration is essentially a two phased migration strategy. The first phase is a migration to the QFX environment where a program can be executed in an emergency situation. The second phase of the migration involves the determination of whether or not the fixed program is to become the permanent version of the program, and the movement, creation, and deletion of modules to implement that decision.

PHASE ONE

- When a request is made to move a program from Emergency in the Test system (EMR) to Quick Fix in Production (QFX), the following actions must take place:

1. The Source must be moved from:

- EMR.SRCMAST.LIBR to QFX.SRCMAST.LIBR.

2. The Source in QFX.SRCMAST.LIBR must be compiled creating:

- A DBRM in QFX.DBRMLIB
- A LOAD in:

♦ QFX.BAT.LOADLIB

or

♦ QFX.CICS.MVS.LOADLIB

or

♦ QFX.CICS.ESA.LOADLIB

3. A Package Bind must be done in production using the QFX DBRM.

4. The PRD DBRM and LOAD are NOT copied to PRE.

5. The QFX DBRM and LOAD are NOT copied to PRD.

PHASE TWO

- If it is determined by the Manager that the current QFX modules are NOT to become the current Production modules, the following steps must be executed:

1. The QFX Source is deleted from QFX.SRCMAST.LIBR

2. The QFX modules are deleted from the QFX libraries in Production

3. The most current Package (QFX version) must be freed.

4. The EMR Source is deleted from EMR.SRCMAST.LIBR

5. The EMR modules are deleted from the EMR libraries in Test.

CCF/MMF MIGRATION

APPENDIX D

- If it is determined by the Manager that the current QFX modules ARE to become the current Production modules, the following steps must be followed:

1. The QFX Source is moved from QFX.SRCMAST.LIBR to STA.SRCMAST.LIBR

2. The QFX modules are deleted from the QFX libraries in Production

3. The most current Package (QFX version) must be freed.

4. The EMR Source is deleted from EMR.SRCMAST.LIBR

5. The normal migration process is initiated WITH THE SOURCE WHICH IS LOCATED IN THE STA.SRCMAST.LIBR (items #1 - #3 of a normal migration are bypassed).

6. The Test Load module must be moved from:

- EMR.BAT.LOADLIB to TPD.BAT.LOADLIB

or

- EMR.CICS.ESA.LOADLIB to TPD.CICS.ESA.LOADLIB

or

- EMR.CICS.MVS.LOADLIB to TPD.CICS.MVS.LOADLIB

7. The Data Base Request Module (DBRM) must be moved from:

- EMR.DBRMLIB to TPD.DBRMLIB

8. A bind must be performed in all test STR(ing) and CAT regions, and in both Training regions, using the concatenation of the CHAIN and TIER DBRM library and the TPD DBRM (CURRENTLY NOT BEING DONE).

-

CCF/MMF MIGRATION

APPENDIX D

EMERGENCY MIGRATION (Phase One)

TEST

EMR	DBRM	SOURCE	LOAD
CREATED	COMPILED	CREATED	
TPD			

PRODUCTION

QFX	DBRM	MOVED	LOAD
CREATED	COMPILED	CREATED	
STA	No Change	No Change	No Change
PRD	No Change		No Change
PRE	No Change		No Change

CCF/MMF MIGRATION

APPENDIX D

EMERGENCY BINDS (Phase One Showing DBRM Library Concatenation)

TEST

UNT

STG

CAT

TRN

PRODUCTION

PRD PRODUCTION

QFX.DBRMLIB

CCF/MMF MIGRATION

APPENDIX D

EMERGENCY MIGRATION (Phase Two)

TEST

EMR	DBRM		LOAD
TPD	DBRM		LOAD
MOVED		MOVED	

PRODUCTION

QFX		SOURCE	
STA	DBRM	MOVED	LOAD
(1)	CREATED	COMPILED	CREATED
PRD	COPIED		COPIED
(3)	from STA 2nd		from STA 2nd
PRE	COPIED		COPIED
(2)	from PRD 1st		from PRD 1st

CCF/MMF MIGRATION

APPENDIX D

SUMMARY of EMERGENCY MIGRATION

PHASE ONE

1. Only the SOURCE is moved to QFX where the precompile and the link create new LOAD and DBRM in the appropriate QFX libraries. These modules are the basis for all executions and binds in the Production environment. The precompile uses the 'VERSION (QFX)' feature. (THIS OPTION IS NOT CURRENTLY USED)

2. The current Production (PRD) LOAD and DBRM and the Previous (PRE) LOAD and DBRM are NOT moved or copied at this time.

PHASE TWO

If a Manager subsequently determines that the changes in QFX are to remain in production:

1. The Source is moved from QFX.SRCMAST.LIBR to STA.SRCMAST.LIBR

2. The modules in QFX are deleted.

3. The LOAD and DBRM in EMR are moved to TPD and become the basis for all executions and binds for all levels in the Test environment.

4. A normal migration is initiated starting with Step #4 of the Normal Migration Process.

CCF/MMF MIGRATION

APPENDIX D

BACKOUT and RECOVERY

A backout of changes, which have gone through the full migration cycle, can be accomplished by deleting the current PRD modules, and using the PRE modules according to the library concatenations.

THIS ACTION, HOWEVER, LEAVES ALL OTHER LIBRARIES IN AN INCONSISTENT STATE WITH THE SOURCE, BINDS, AND EXECUTION MODULES IN BOTH TEST AND PRODUCTION.

- The following actions must be taken, after a full migration, to place all libraries back into a consistent state.

1. The -1 SOURCE archive level in STA(ge) must be selected and saved as the current level. This action will force the backout version of the program source to be demoted to the -1 version.

2. The LOAD MODULE in PRE must be copied to PRD.

3. The LOAD MODULE in PRE must be copied to STA.

4. The DBRM in PRE must be copied to PRD.

5. The DBRM in PRE must be copied to STA.

6. No Bind is necessary (the previous Package Bind will match the PRE(vious) (and now current (PRD)) Load Module).

7. The SOURCE must be checked out from PRD into EMR.

8. The SOURCE must be compiled in EMR.

9. The LOAD MODULE and the DBRM must be moved to the proper TPD libraries.

10. A Package Bind must be done in all test STR(ing) and CAT regions, and in both training regions, using the concatenation of the CHAIN and TIER DBRM library and the TPD DBRM library.

11. The SOURCE, LOAD, and DBRM must be deleted from the EMR libraries.

- If a program has been migrated to STA(ge), but has not been moved to Production, the following actions must be taken to place all libraries back into a consistent state.

1. The -1 SOURCE archive level in STA(ge) must be selected and saved as the current level. This action will force the backout version of the program source to be demoted to the -1 version.

2. If the SOURCE has been compiled:

- The LOAD in PRD must be moved to STA(ge)

- The DBRM in PRD must be moved to STA(ge)

CCF/MMF MIGRATION

APPENDIX D

3. No Bind is necessary (the previous Package Bind will still match the current PRD Load Module).

4. If a Bind has been accomplished:

- The most current Package must be freed.

5. If the program is going to be promoted again, no further action is required.

6. If the program is NOT going to be promoted again, the following actions are required:

- The SOURCE must be checked out from PRD into EMR.
- The SOURCE must be compiled in EMR.
- The LOAD MODULE and the DBRM must be moved to the proper TPD libraries.
- A Package Bind must be done in all test STR(ing) and CAT regions, and in both training regions, using the concatenation of the CHAIN and TIER DBRM library and the TPD DBRM library.
- The SOURCE, LOAD, and DBRM must be deleted from the EMR libraries.

CCF/MMF MIGRATION

APPENDIX D

BACKOUT

TEST

EMR	DBRM	SOURCE	LOAD
CREATED	COMPILED	CREATED	
TPD	DBRM		DBRM
MOVED		MOVED	

PRODUCTION

STA		SELECT –1	
COPIED	SAVE	COPIED	
(3)	from PRE 3rd		from PRE 3rd
PRD	COPIED		COPIED
(3)	from PRE 2nd		from PRE 2nd
PRE	No Change		No Change

CCF/MMF MIGRATION

APPENDIX D

TRAINING MIGRATION

- When training is scheduled with the T/A group, a special AWP job is run. This job reinitializes the DB2 tables and rebinds all previously identified programs using the DBRM located in the TPD.DBRMLIB.

- If Training Region 3 (ESATRN3 or MVSTRN3) is to be used, and programs are to be used from the CAT environment, a request to bind the program(s) from either CAT Chain A or CAT Chain B must accompany the scheduling request.

DB2 MIGRATION/CONVERSION CHECKLIST

APPENDIX E

INTRODUCTION

The following document has been created to be used as a guide for the migration or conversion of programs or files which use DB2. While it is a guide, and can be modified as may be appropriate, the document is intended to be used to insure that all significant and important issues and concerns are addressed and considered in a typical DB2 migration or conversion.

This document is designed to be used by all agencies which require support from the CDP Data Base Management group. Based upon whether it is a (1) Simple Migration, (2) Complex Migration, or (3) Complex Conversion (the definitions are included in the Appendix A), separate sections have been prepared to highlight the information and actions needed to encourage a successful project implementation.

A second section, titled 'Test System Migration Worksheet' has been created for systems which use IM Services' CCF/MMF migration process. Agencies which use their own methodology and testing scheme should develop their own test migration scheme. See Appendix B for a sample of this checklist.

In all cases, the document may be modified to meet special needs which may arise in a project. When modified or changed, those changes need to be communicated to all concerned to eliminate the chance of misunderstanding. When items are intentionally not included in a plan, explicit notation of the exclusion should be made.

As procedures are modified, or as additional subjects are found to be necessary or helpful, changes will be made to this basic document. In the event portions are found to be redundant or unnecessary, they can be nominated for removal.

APPENDIX E

SIMPLE MIGRATION or CONVERSION

This section should be used for the movement of new programs and/or files which are involved in 'READ ONLY' and/or reporting activity, but which do not update permanent data files.

PLANNING

Applications Development:

- SYSTEM PLAN (Collection) identified ____
- New program(s)/file(s)/control(s) identified ____
- CICS Support Notified (PPT, transaction) ____
- DBM Notified (RCT) ____
- New file(s) defined ____
- GDG's defined ____
- DB2 Control Cards created and checked ____
- Operations documentation (APCDOC) complete ____
- JCL approved ____
- JOB schedule conflicts checked and resolved ____
- Dependencies and time of day ____
- Security authorizations identified ____

DOR Data Administration:

- Attach RACF groups to Transaction Id's ____
- Security authorizations applied ____

Data Base Management:

- SYSTEM PLAN created ____
- DB Support Notified (RCT) ____
- Plan Execution Authority granted ____
- New program(s) identified ____

- MVS Ids and authorizations identified ____

APPENDIX E

IMPLEMENTATION

Applications Development:

- Manager has reviewed and confirmed the completion

of all planning items ____

- Managers approval for Implementation ____
- Tests Reviewed and Approved ____
- Users Notified of Schedule ____

Data Base Management:

- Manager has reviewed and confirmed the completion

of all planning items ____

- Managers approval for Implementation ____
- Execute Bind ____
- PLAN_TABLE Review(s) ____
- Plan or Package Execution Authorization ____
- DDF Table entries made ____

POST-IMPLEMENTATION

Applications Development:

- Programs tested and functioning as expected ____

Data Base Management:

- DB2 PLAN active ____

APPENDIX E

COMPLEX MIGRATION

This section should be used for the movement of new programs and/or files which are involved in the update of new permanent data files.

PLANNING

Applications Development:

- SYSTEM PLAN (Collection) identified ____
- New program(s)/file(s)/control(s) identified ____
- CICS Support Notified (PPT, transaction) ____
- DBM Notified (RCT) ____
- New file(s) defined ____
- GDG's defined ____
- DB2 Control Cards created and checked ____
- Operations documentation (APCDOC) complete ____
- JCL approved ____
- JOB schedule conflicts checked and resolved ____
- Dependencies and time of day ____
- New DB2 Table Object(s) ____
- Tables Defined and Normalized ____
- New View(s) required ____
- Columns (defaults) Reviewed ____
- Relationships Verified ____
- Delete Rules Verified ____
- Identify Audit and Data Capture Requirements ____
- Table Authorization requirements defined ____
- DBM Notified of authorization requirements ____

- Security authorizations identified ____

APPENDIX E

DOR Data Administration:

- Attach RACF groups to Transaction Id's ____
- Security authorizations applied ____
- Table Authorizations reviewed ____
- Utilities JCL Generated ____
- DB2 Control Cards created and checked ____
- Image copies ____
- Platinum Stats ____
- Modify Recovery ____

Data Base Management:

- SYSTEM PLAN created ____
- DB Support Notified (RCT) ____
- Plan Execution Authority granted ____
- New program(s) identified ____
- New DB2 Table Object(s) ____
- Tables Defined and Normalized ____
- Columns (defaults) Reviewed ____
- Relationships Verified ____
- Delete Rules Verified ____
- Indexes Reviewed ____
- Table Authorizations Reviewed ____
- Audit and Data Capture Requirements Reviewed ____
- DDF Table entries made ____
- Space Calculations Completed ____
- STOGROUP and Volumes Reviewed ____
- DDL Generated ____

- Utilities JCL Generated ____
- DB2 Control Cards created and checked ____

DB2 MIGRATION/CONVERSION CHECKLIST

APPENDIX E

- Image copies ____
- Platinum Stats ____
- Modify Recovery ____
- Log Analyzer ____
- APCDOC complete ____
- JCL approved ____
- New file(s) defined ____
- GDG's defined ____
- JOB schedule conflicts checked and resolved ____
- Dependencies and time of day ____

IMPLEMENTATION

Applications Development:

- Contact List Prepared and sent to DBM and Operations ____
- Manager has reviewed and confirmed the completion

of all planning items ____

- Managers approval for Implementation ____
- Tests Reviewed and Approved ____
- Users Notified of Schedule ____
- Populate new tables and Initial Load jobs ____

Data Base Management:

- Manager has reviewed and confirmed the completion

of all planning items ____

- New DB2 Table Object(s) ____
- STOGROUP determined ____
- Data base created ____
- Tablespace(s) created ____
- Table(s) created ____
- Index(es) created ____

APPENDIX E

- Tables altered for Referential Integrity ____
- DDL Executed ____
- Data Base Authorizations for Utilities ____
- DDF Table entries ____
- Table Authorizations ____
- Utilities JCL Scheduled ____
- Application Initial Load(s) Complete ____
- Execute Bind ____
- PLAN_TABLE Review ____
- Image copies and other utilities ____

POST-IMPLEMENTATION

Applications Development:

- Programs tested and functioning as expected ____
- Data verified ____

Data Base Management:

- DB2 PLAN active ____
- DB2 Data Base and Tables available (no flags) ____
- Tables Structures verified ____
- Applications Notified ____

DB2 MIGRATION/CONVERSION CHECKLIST

APPENDIX E

COMPLEX CONVERSION

This section should be used for the movement of changed and new programs and/or changed and new files which are involved in the update or the storage of new or changed permanent data.

PLANNING

Applications Development:

- SYSTEM PLAN (Collection) identified ____
- New program(s)/file(s)/control(s) identified ____
- CICS Support Notified (PPT, transaction) ____
- DBM Notified (RCT) ____
- New file(s) defined ____
- GDG's defined ____
- DB2 Control Cards created and checked ____
- Operations documentation (APCDOC) complete ____
- Change to Package Bind ____
- Utilities Changes identified ____
- JCL approved ____
- JOB schedule conflicts checked and resolved ____
- Dependencies and time of day ____
- New DB2 Table Object(s) ____
- Tables Defined and Normalized ____
- New View(s) required ____
- Columns (defaults) Reviewed ____
- Relationships Verified ____
- Delete Rules Verified ____
- Table Authorization requirements defined ____

- DBM Notified of authorization requirements ____
- Security Changes identified ____

APPENDIX E

- Old DB2 Table Object(s) ____
- Table changes identified and documented ____
- View(s) changes required ____
- Columns (defaults) Reviewed ____
- Relationships Verified ____
- Delete Rules Verified ____
- DBM Notified of changes by object ____
- DBM Notified of authorization changes ____
- Backout Contingency Plan complete and reviewed ____
- Time selected for changes ____
- Business Users Contacted ____

DOR Data Administration:

- Attach RACF groups to Transaction Id's ____
- Security authorizations applied ____
- Table Authorizations reviewed ____
- Utilities JCL Generated ____
- DB2 Control Cards created and checked ____
- Image copies ____
- Platinum Stats ____
- Modify Recovery ____

Data Base Management:

- SYSTEM PLAN created ____
- New DB2 Table Object(s) ____
- Tables Defined and Normalized ____
- Columns (defaults) Reviewed ____

- Relationships Verified ____
- Delete Rules Verified ____

APPENDIX E

- Indexes Reviewed ____
- New View(s) required ____
- New Authorizations required ____
- JCL Changes ____
- Utilities Changes ____
- Source for DDL Generation Confirmed ____
- Table Authorizations defined ____
- DDF Table entries Completed ____
- Space Calculations Completed ____
- STOGROUP and Volumes Reviewed ____
- DDL Generated ____
- Utilities JCL Generated ____
- Image copies ____
- PLATINUM Stats ____
- Modify Recovery ____
- Log Analyzer ____
- DB2 Control Cards created and checked ____
- APCDOC complete ____
- JCL approved ____
- GDG's defined ____
- JOB schedule conflicts checked and resolved ____
- Dependencies and time of day ____
- Changed DB2 Table Object(s) ____
- Run Migrator to generate old DDL ____
- PLATINUM BP Statements Generated ____

- Unloads ____
- Drops ____
- Creates ____
- Authorizations ____
- Synonyms ____

APPENDIX E

- ♦ Reloads ____
- ♦ Check Utilities ____
- ♦ Copy Utilities ____
- ♦ Rebinds ____
- ▪ Space Recalculated ____
- ▪ STOGROUP and Volumes Verified ____
- ▪ Utilities JCL Generated ____
- ▪ APCDOC complete ____
- ▪ JCL approved ____
- ▪ GDG's defined ____
- ▪ Current DB2 Control Cards identified ____
- ▪ JOB schedule conflicts checked and resolved ____
- ▪ Dependencies and time of day ____
- ▪ List of contacts (locations and telephone numbers) ____
- ▪ Operations ____
- ▪ CICS ____
- ▪ DBM ____
- ▪ Applications Development ____
- ▪ User ____

IMPLEMENTATION

Applications Development:

- ▪ Contact List Prepared and sent to DBM and Operations ____
- ▪ Manager has reviewed and confirmed the completion

of all planning items ____

- Managers approval for Implementation ____
- Tests Reviewed and Approved ____
- Users Notified of Schedule ____
- New DB2 Table Object(s) ____

APPENDIX E

- Application Initial Load(s) ____
- Old DB2 Table Object(s) ____
- Application Conversion Program(s) Run ____

Data Base Management:

- Manager has reviewed and confirmed the completion

of all planning items ____

- New DB2 Table Object(s) ____
- STOGROUP determined ____
- Data base created ____
- Tablespace(s) created ____
- Table(s) created ____
- Index(es) created ____
- Exception Table(s) created and/or altered ____
- Tables altered for Referential Integrity ____
- DDL Executed ____
- Data Base Authorizations for Utilities ____
- Table Authorizations ____
- Utilities JCL Scheduled ____
- Old DB2 Table Object(s) ____
- Unload Tables using DSNTIAUZ ____
- PLATINUM BP Executed ____
- Application Initial Load(s) Complete ____
- Execute Bind ____
- PLAN_TABLE Review ____
- Conversion Program(s) Complete ____
- New RI Required ____

- Image copies and other utilities ____
- DB2 Control cards modified ____

DB2 MIGRATION/CONVERSION CHECKLIST

APPENDIX E

POST-IMPLEMENTATION

Applications Development:

- Programs tested and functioning as expected ____
- Data verified ____

Data Base Management:

- DB2 PLAN active ____
- DB2 Data Base and Tables available (no flags) ____
- Tables Structures verified ____
- Data verified (Utility Row Counts) ____
- Applications Notified ____
- Verify Table Authorizations ____
- Verify Utilities JCL changed and moved ____

DB2 MIGRATION/CONVERSION CHECKLIST

APPENDIX E

Appendix A

DEFINITIONS

CHANGE TYPES

1. MIGRATION - the movement of programs or files, IMS data bases, or DB2 table objects, from test to production, which are new, and which do not effect any other program, file, IMS data base, or DB2 table object already in production.

2. CONVERSION - - the movement of programs or files, IMS data bases, or DB2 table objects, from test to production, which are either new or changed, and which do effect other programs, files, IMS data bases or DB2 table objects which are already in production.

COMPLEXITY

1. SIMPLE MIGRATION or CONVERSION - the movement of new programs and/or files which are involved in 'READ ONLY' and/or reporting activity, but which do not update permanent data files.

2. COMPLEX MIGRATION - the movement of new programs and/or files, which are involved in the update of new, permanent data files.

3. COMPLEX CONVERSION - the movement of changed and new programs and/or changed and new files which are involved in the update or the storage of new or changed permanent data.

MANAGERS

1. MANAGER - a person who writes a performance evaluation or review on another person.

2. SENIOR MANAGER - a person who writes a performance evaluation or review on another manager.

DB2 MIGRATION/CONVERSION CHECKLIST

APPENDIX E

COMMENTS and NOTES

DB2 MIGRATION/CONVERSION CHECKLIST

APPENDIX E

TEST SYSTEM MIGRATION WORKSHEET

PATH 1	PATH 2	PATH 3
_ ____UDB01	_ ____UDB02	_ ____UDB03
_ ____ADU1	_ ____ADU2	_ ____ADU3
_ ____APU1	_ ____APU2	_ ____APU3
_ AUTHS – TABLE	_ AUTHS – TABLE	_ AUTHS - TABLE
UNT _ ____ONLU1	_ ____ONLU2	_ ____ONLU3
_ ____BATU1	_ ____BATU2	_ ____BATU3
_ AUTHS – PLAN	_ AUTHS – PLAN	_ AUTHS - PLAN
_ BINDS (ONL/BAT)	_ BINDS	_ BINDS
_ RCT (MVSUNT1)	_ RCT (MVSUNT2)	_ RCT (MVSUNT3)
_ RCT (ESAUNT1)	_ RCT (ESAUNT2)	_ RCT (ESAUNT3)
_ DB2 CNTL CARD	_ DB2 CNTL CARD	_ DB2 CNTL CARD
_ PLAN_TABLE	_ PLAN_TABLE	_ PLAN_TABLE
_ DCLGENS	_ DCLGENS	_ DCLGENS
_ CCF_ACRONYM entry	_ CCF_ACRONYM entry	_ CCF_ACRONYM entry
_ ____SDB01	_ ____SDB02	_ ____SDB03
_ ____ADS1	_ ____ADS2	_ ____ADS3
_ ____APS1	_ ____APS2	_ ____APS3
_ AUTHS – TABLE	_ AUTHS – TABLE	_ AUTHS - TABLE
STG _ ____ONLS1	_ ____ONLS2	_ ____ONLS3
_ ____BATS1	_ ____BATS2	_ ____BATS3
_ AUTHS – PLAN	_ AUTHS – PLAN	_ AUTHS - PLAN

_ BINDS (ONL/BAT) _ BINDS _ BINDS

_ RCT (MVSSTG1) _ RCT (MVSSTG2) _ RCT (MVSSTG3)

_ RCT (ESASTG1) _ RCT (ESASTG2) _ RCT (ESASTG3)

_ DB2 CNTL CARD _ DB2 CNTL CARD _ DB2 CNTL CARD

_ PLAN_TABLE _ PLAN_TABLE _ PLAN_TABLE

_ DCLGENS _ DCLGENS _ DCLGENS

_ CCF_ACRONYM entry _ CCF_ACRONYM entry _ CCF_ACRONYM entry

==========

_ ____CDB01 _ ____CDB02 _ ____CDB03

_ ____ADC1 _ ____ADC2 _ ____ADC3

_ ____APC1 _ ____APC2 _ ____APC3

_ AUTHS – TABLE _ AUTHS – TABLE _ AUTHS - TABLE

CAT _ ____ONLC1 _ ____ONLC2 _ ____ONLC3

_ ____BATC1 _ ____BATC2 _ ____BATC3

_ AUTHS – PLAN _ AUTHS – PLAN _ AUTHS - PLAN

_ BINDS (ONL/BAT) _ BINDS _ BINDS

_ RCT (MVSCAT1) _ RCT (MVSCAT2) _ RCT (MVSCAT3)

_ RCT (ESACAT1) _ RCT (ESACAT2) _ RCT (ESACAT3)

_ DB2 CNTL CARD _ DB2 CNTL CARD _ DB2 CNTL CARD

_ PLAN_TABLE _ PLAN_TABLE _ PLAN_TABLE

_ DCLGENS _ DCLGENS _ DCLGENS

_ CCF_ACRONYM entry _ CCF_ACRONYM entry _ CCF_ACRONYM entry

==========

DB2 MIGRATION/CONVERSION CHECKLIST

APPENDIX E

FREE PLANS:

DROP DATABASE:

Delete RACF Groups:

DB2 MIGRATION/CONVERSION CHECKLIST

APPENDIX E

. (Page 7)

+++ OL end-tag found outside List and ignored. (Page 7)